

# Informix Backup and Restore - The Bare Minimum

Basic Informix backup and restore is pretty simple and straight forward in recent versions of Informix.

I'm going to assume you don't have or don't want to use a tape storage device and a Storage Manager. I'm assuming you just want to backup your data to the filesystem and you'll do the due diligence of moving these backup files off site or at least to a different server/disks to provide maximum recoverability. For this we will use the ontape Informix utility and the (relatively) new feature of automatically backing up and restoring from a directory.

There are 3 kinds of backups, System Backups, Logical Log Backups and Critical File Backups.

## System Backups

A System Backup is a direct copy of the Informix pages that make up your engine. This is done with the engine online and with no impact or blocking of users other than the additional I/O strain put on the system to read each page from the engine and write out each page to disk.

There are 3 levels of System Backup that can be taken:

- A Level 0 backs up every page
- A Level 1 backs up every page that has changed since the last Level 0 backup
- A Level 2 backs up every page that has changed since the last Level 1 backup

If you use multiple backup levels, when it comes time to restore you would first restore the last Level 0 then apply the most recent Level 1 backup (if one exists) after the Level 0 and finally apply the most recent Level 2 backup (if one exists) after the Level 1 backup is restored.

Using Level 1 and Level 2 backups can reduce the size of your backups taken and the I/O resources required to write backups to disk (less data to write, we still have to read all of the data), but we're here for the bare minimum so I'm just going to use Level 0 backups and Logical Log backups to keep our data safe.

Using a System Backup to restore data is called a Physical Restore.

## Logical Log Backups

Each logical log should be backed up when it becomes full. Not only because it will help you recover lost data but because Informix will BLOCK if it needs to reuse a logical log that has not been backed up (remember the logical logs are used in a cyclical fashion) and it will continue to block until the needed logical log is backed

up. Informix does this because it assumes data recoverability is more important than engine availability. Whether you agree with this philosophy or not it is the way the engine works, so you will need to backup your logical logs to prevent the engine from blocking.

There is a work around, but it is not recommended. You can set LTAPEDEV to /dev/null in the ONCONFIG file which will disable logical log backups and automatically mark a logical log as backed up when it stops being the current logical log. If you do this your recovery from a failure will be limited to the data stored in your Level 0, 1 and 2 System Backups.

The logical log backups are used during a restore to replay transactions that have occurred since the last System Backup. After the Physical Restore of a Level 0 (and optionally a Level 1 and 2) backup you perform a Logical Restore of the backed up logical logs.

### **Logical Log Salvage**

If your engine crashes and needs to be restored there will be at least one logical log that has not been backed up yet, the current logical log. It would be real nice to be able to backup this logical log before we try and restore so it can be in the list of logical log backups to restore our engine to the most up to date version before the crash. Well the engineers that implemented Informix were real smart and gave us this ability. Before a Physical Restore is attempted you will be asked if you want to backup the logical logs and salvage them before we overwrite them with a restore. This works great as long as the disk that holds the logical logs isn't the thing causing you to restore.

### **Critical File Backups**

There are some files that you should backup each time you take a System Backup, these files will help you recreate your Informix install in the event that the filesystem that holds your Informix install and config files becomes unavailable

- \$INFORMIXDIR/etc/\$ONCONFIG
- \$INFORMIXSQLHOSTS or \$INFORMIXDIR/etc/sqlhosts
- \$INFORMIXDIR/etc/oncfg\_\*
- onstat -d output, to give you a listing of the chunk paths if you need to recreate them

Unfortunately there is no Informix utility for this, you should write a script to tar.gz these files and move them somewhere safe each time a System Backup is taken. Something like this should get you pointed in the right direction.

```
informix> onstat -d > /home/informix/tmp/onstat.d.out
informix> tar -czf /home/informix/backup/files/criticalfiles.$(date
"+%Y%m%d.%H%M").tar.gz $INFORMIXDIR/etc/$ONCONFIG
$INFORMIXSQLHOSTS $INFORMIXDIR/etc/oncfg_*
```

/home/informix/tmp/onstat.d.out

## **Taking a System Backup**

We will be using the Informix utility `ontape` and its ability to backup to a directory for all of our System and Logical Log backup needs.

To enable `ontape` System Backups to a directory, modify the `ONCONFIG` parameter `TAPEDEV` to a directory name that is owned by user `informix` and group `informix` and has read, write, execute permissions for user and group. This is where the System Backups will live.

```
informix> mkdir /home/informix/backup
informix> mkdir /home/informix/backup/system
informix> chmod 770 /home/informix/backup/system
informix> vi $INFORMIXDIR/etc/$ONCONFIG
```

`TAPEDEV /home/informix/backup/system`

The most recent System Backup will have a name of `DBSERVERNAME_SERVERNUM_L[0, 1 or 2]`. A Level 0 backup for our dev engine will be `/home/informix/backup/system/dev_0_L0`

Each time you execute a new System Backup the old backup is renamed with a timestamp inserted between `SERVERNUM` and `L[0, 1 or 2]`

A listing of all the backups of my dev engine looks like this

```
dev_0_20100723_133319_L0
dev_0_20100725_135325_L0
dev_0_L0
```

To actually take a Level 0 backup

```
informix> ontape -s -L 0 -d
```

`-s` tells `ontape` we want a System Backup, `-L 0` asks for a Level 0 (`-L 1` and `-L 2` for Level 1 or 2 backups) and `-d` says we're backing up to a directory and prevents `ontape` from prompting us for any input.

I recommend executing this in a script that also backs up the critical files via cron or some other scheduling mechanism, simply remembering to take a backup every day or week and kicking it off manually isn't good enough.

## Taking Logical Log Backups

There are multiple ways to use ontape to backup logical logs to a directory, but I think the best way is to use the ALARMPROGRAM, a script called by Informix when events occur, to back them up as they fill.

Your ONCONFIG ALARMPROGRAM parameter should already be set to \$INFORMIXDIR/etc/alarmprogram.sh, all we have to do slightly modify this IBM supplied script and set LTAPEDEV in the ONCONFIG to a directory.

```
informix> mkdir /home/informix/backup/llog
informix> chmod 770 /home/informix/backup/llog
informix> vi $INFORMIXDIR/etc/$ONCONFIG
```

```
LTAPEDEV /home/informix/backup/llog
```

```
informix> vi $INFORMIXDIR/etc/alarmprogram.sh
```

```
# line 31, change flag from N to Y
BACKUPLOGS=Y
```

```
# line 62, change onbar -b -l to ontape -a -d
BACKUP_CMD="ontape -a -d"
```

That's is, you shouldn't have to do anything else. Each time a logical log fills up alarmprogram.sh will be called and any unbacked up logical logs will be backed up to your backup/llog directory.

## Restoring an Engine

I will focus on a Cold Restore, which requires the engine to be offline. Presumably because something terrible happened and you need to restore to recover from a server or disk problem.

Using the output from the onstat -d you saved during your last system backup, verify that the files for your chunks exist. If you replaced a failed drive or are restoring to a different server they may not exist, in this case you can use the touch command to recreate a 0 byte file placeholder with the appropriate ownership and permissions. The restore will do the rest.

Using the ontape utility, invoke the restore command with the -d directory option. When -d is used you are not prompted for any decisions, ontape figures out what to do based on what is in the backup directories specified in TAPEDEV and LTAPEDEV.

informix> ontape -r  
Restore is using file /home/informix/backup/data/dev\_0\_L0 ...

#### Archive Tape Information

Tape type: Archive Backup Tape  
Online version: IBM Informix Dynamic Server Version 14.10.UC7IE  
Archive date: Sun Jul 25 16:24:17 2020  
User id: informix  
Terminal id: /dev/pts/0  
Archive level: 0  
Tape device: /home/informix/backup/data/  
Tape blocksize (in k): 32  
Tape size (in k): system defined for directory  
Tape number in series: 1

Spaces to restore:1

[rootdbs ]

2

[llogdbs01

]

#### Archive Information

IBM Informix Dynamic Server Copyright 2001, 2010 IBM Corporation.

Initialization Time 07/23/2020 15:50:16

System Page Size 2048

Version 18

Index Page Logging OFF

Archive CheckPoint Time 07/25/2020 16:13:05

#### Dbspaces

number	flags	fchunk	nchunks	flags	owner	name
1	40001	1	1	N B	informix	rootdbs
2	40001	2	1	N B	informix	llogdbs01

#### Chunks

chk/dbs	offset	size	free	bpages	flags	pathname
1 1 0	1048576	518117			PO-B	/home/informix/chunks/ROOTDBS.01

2 2 0 1048576 425969 PO-B /home/informix/chunks/LLOGDBS01.01

Continue restore? (y/n)y

Do you want to back up the logs? (y/n)y

File created: /home/informix/backup/llog/dev\_0\_Log0000000040

Log salvage is complete, continuing restore of archive.

Restore a level 1 archive (y/n) y

Ready for level 1 tape

Restore is using file /home/informix/backup/data/dev\_0\_L1 ...

File /home/informix/backup/data/dev\_0\_L1 not found, continuing ...

Do you want to restore log tapes? (y/n)y

Roll forward should start with log number 39

Restore is using file /home/informix/backup/llog/dev\_0\_Log0000000039 ...

Rollforward log file /home/informix/backup/llog/dev\_0\_Log0000000039 ...

Rollforward log file /home/informix/backup/llog/dev\_0\_Log0000000040 ...

Program over.

After ontape -r -d completes the engine will be in Quiescent (Single User) mode, resolve this by executing onmode -m to switch to Multi-User Mode and finalize the restore.

informix> onmode -m

## Keeping the Size of Backups In Check

If you would like to reduce the amount of space these backups are taking you're probably thinking about compressing them with gzip or something similar and decompressing the required files before a restore. Well those Informix guys and gals did it again by adding 2 ONCONFIG parameters that allow you to automatically compress and decompress these backup files on the fly via backup and restore filters.

If I would like to use gzip to compress my backups and gunzip to decompress I would modify the BACKUP\_FILTER and RESTORE\_FILTER ONCONFIG parameters. I will also be required to change LTAPESIZE from the default of 0 (unlimited) to something not 0. Not sure exactly why this is required, but the engine complains if LTAPESIZE is 0.

informix> vi \$INFORMIXDIR/etc/\$ONCONFIG

LTAPESIZE 2097152

BACKUP\_FILTER /bin/gzip

RESTORE\_FILTER /bin/gunzip

## Informix HDR

If I had to guess, I would say that most production database engines utilize RAID technology to protect against the inevitable disk failure and the ones that don't probably should. Disk is cheap and the revenue saved by avoiding an extended outage can be enough to pay for disk mirroring many times over.

If I had to guess again, I would say that not nearly enough production database engines utilize High Availability Data Replication (HDR) to protect against the inevitable server failure. Why is this? Servers can fail too. Sure, servers are more expensive than disks and sure the MTTF is longer than disks but the money lost during an extended outage that could have been avoided with HDR is probably going to be more than the cost of implementing an HDR solution.

HDR continuously replicates the changes made to a Primary server to a Secondary server that can be quickly converted to a Primary if the original Primary fails. As an added bonus, the Secondary server can be used for reads and writes allowing you to make use of this hardware to improve performance instead of letting it sit there idle. You could also implement multiple Remote Standalone Secondary (RSS) or Shared Disk Secondary (SDS) servers to create a grid if your Informix Edition supports this. I'm going to focus on a single HDR Secondary which is available for no cost in Innovator-C.

As with most Informix features, HDR is incredibly easy to configure and does not require much administration.

### Get Yourself Some More Hardware

To enable HDR you will need another server. This server should be identical to the Primary server. The Secondary server doesn't have to be identical in every way, but if you expect it to take over during a failure you're going to want the same amount of memory, CPUs, etc. to ensure it can handle the load. Here is what is required of servers participating in HDR:

Both servers must run the same Informix version

- Both servers must be able to run the same Informix executable. Ubuntu and Red Hat run the same Informix executable; HP/UX and Red Hat do not. Why would you ever want to do this anyway?
- Both servers must have network capabilities
- The Secondary server must have at least as much disk space for dbspaces as the Primary. The dbspace chunk types (cooked or raw) do not have to be identical
- Dbspace chunk path names must be identical, symbolic links can help here

- Not really a hardware requirement but any databases you want replicated must be logged. Unbuffered logging is preferred

## Install and Configure Informix on the New Server

Follow the steps from [Installing Innovator-C on Linux](#) on a new server named dev02. Create 0 byte files with the touch command to mirror the dbspace chunks on the primary server.

```
informix@dev02> mkdir /home/informix/chunks
informix@dev02> touch /home/informix/chunks/ROOTDBS.01
informix@dev02> touch /home/informix/chunks/LLOGDBS01.01
informix@dev02> touch /home/informix/chunks/DATADBS01.01
informix@dev02> touch /home/informix/chunks/DATADBS01.02
informix@dev02> chmod 660 /home/informix/chunks/*
```

Copy the /etc/profile.d/informix.sh file from the Primary to the Secondary and change INFORMIXSERVER

```
root@dev02> scp dev:/etc/profile.d/informix.sh
/etc/profile.d/informix.sh
root@dev02> vi /etc/profile.d/informix.sh

export INFORMIXSERVER=dev02
```

Copy the ONCONFIG file from the Primary to the Secondary and change DBSERVERNAME and add a DBSERVERALIASES to both ONCONFIGs that will be used exclusively for HDR.

```
informix@dev> vi $INFORMIXDIR/etc/$ONCONFIG

DBSERVERALIASES dev_hdr

informix@dev02> scp dev:/opt/informix/etc/onconfig.dev
$INFORMIXDIR/etc/$ONCONFIG
informix@dev02> vi $INFORMIXDIR/etc/$ONCONFIG

DBSERVERNAME dev02
DBSERVERALIASES dev02_hdr
```

Do we need a dedicated connection for HDR? No, but I feel doing so gives me two advantages

- I can put HDR traffic on a separate network if I want
- Both HDR servers must trust each other, I can use the more secure \$INFORMIXDIR/etc/hosts.equiv to accomplish this if HDR runs on a dedicated port

If you would like to allow insert, update and deletes to take place on the Secondary server set the UPDATABLE\_SECONDARY ONCONFIG parameter on both servers to a number between 1 and CPUVPs \* 2 to configure the number of threads for transmitting updates from the Secondary to the Primary.



```
informix> vi $INFORMIXDIR/etc/$ONCONFIG
```

```
UPDATABLE_SECONDARY 2
```

Add a new port to /etc/services on both servers for HDR.

```
root> vi /etc/services
```

```
idshdr01          1528/tcp          # Informix HDR
```

Modify the sqlhosts file on both the Primary and the Secondary so they both contain connectivity information for both servers. Use the s=6 security option for the HDR ports to indicate that only Replication traffic is allowed on these ports giving us the ability to use \$INFORMIXDIR/etc/hosts.equiv to establish trusts.

```
informix> vi $INFORMIXSQLHOSTS
```

```
# dev
dev          onsoctcp          dev          idstcp01
dev_hdr      onsoctcp          dev          idshdr01
s=6

# dev02
dev02        onsoctcp          dev02        idstc
p01
dev02_hdr    onsoctcp          dev02        idshd
r01          s=6
```

Bounce the Primary server for ONCONFIG changes to take effect.

### Create or Modify an Existing hosts.equiv Files

The hosts.equiv file will contain the hostname of each server that is allowed to make a trusted connection. You must also change the permissions of the file so only the informix user can write to it.

```
informix@dev> vi $INFORMIXDIR/etc/hosts.equiv
```

```
dev02
```

```
informix@dev> chmod 640 $INFORMIXDIR/etc/hosts.equiv
```

```
informix@dev02> vi $INFORMIXDIR/etc/hosts.equiv
```

```
dev
```

```
informix@dev02> chmod 640 $INFORMIXDIR/etc/hosts.equiv
```

Note: Later when we start HDR if you see messages in your online.log (onstat -m output) that look like this:

```
12:12:16 listener-thread: err = -956: oserr = 0: errstr =
informix@dev02.prod.informix-dba.com[dev02]: Client host or
user informix@dev02.prod.informix-dba.com[dev02] is not
trusted by the server.
```

then need you to put the full hostname, dev02.prod.informix-dba.com, in hosts.equiv

## Restore Secondary Server Using a Backup from the Primary

The first step in actually starting HDR is to perform a physical restore of the Primary to the Secondary. After this is complete we will start HDR and Informix will automatically sync the Secondary with the Primary by processing the logical log records that have been written since the Primary's backup was taken.

One of my favorite Informix features is ontape to STDIO, you can use this feature to simultaneously take a Level 0 backup of your Primary, ship the data over the network and pipe it directly into a physical restore on the Secondary. This is a lot easier than performing an Imported Restore. Like to see it? Here it goes.

```
informix@dev> ontape -s -L 0 -F -t STDIO | ssh informix@dev02  
". /etc/profile.d/informix.sh; ontape -p -t STDIO"
```

While this is running, you can use onstat -D on both servers to see the reading of pages on the Primary and the writing of pages on the Secondary in parallel. After the backup and restore completes the Secondary server will be in Fast Recovery mode.

```
informix@dev02> onstat -m  
  
IBM Informix Dynamic Server Version 14.10.UC7IE -- Fast  
Recovery -- Up 00:00:40 -- 1164976 Kbytes  
  
Message Log File: /opt/informix-ids-14.10.UC7IE/tmp/online.log  
13:38:11 Maximum server connections 0  
13:38:11 Checkpoint Statistics - Avg. Txn Block Time 0.000, #  
Txns blocked 0, Plog used 0, Llog used 0  
  
13:38:11 Checkpoint Completed: duration was 0 seconds.  
13:38:11 Tue Aug 3 - logunig 10, logpos 0x1816018,  
timestamp: 0x4a722 Interval: 721  
  
13:38:11 Maximum server connections 0  
13:38:11 Checkpoint Statistics - Avg. Txn Block Time 0.000, #  
Txns blocked 0, Plog used 0, Llog used 0  
  
13:38:11 Checkpoint Completed: duration was 0 seconds.  
13:38:11 Tue Aug 3 - logunig 10, logpos 0x1816018,  
timestamp: 0x4a728 Interval: 722  
  
13:38:11 Maximum server connections 0  
13:38:11 Checkpoint Statistics - Avg. Txn Block Time 0.000, #  
Txns blocked 0, Plog used 0, Llog used 0  
  
13:38:12 Physical Restore of rootdbs, llogdbs01, datadbs01  
Completed.  
13:38:12 Checkpoint Completed: duration was 0 seconds.  
13:38:12 Tue Aug 3 - logunig 10, logpos 0x1816018,  
timestamp: 0x4a739 Interval: 722  
  
13:38:12 Maximum server connections 0
```

and you are ready to start HDR.

## Starting HDR

Start HDR on the Primary with the onmode -d primary command. In this command you will tell Informix that this is a Primary HDR server and the Secondary is dev02.

```
informix@dev> onmode -d primary dev02
```

Start HDR on the Secondary with the onmode -d secondary command. This will tell Informix that this is a Secondary HDR server and the Primary is dev.

```
informix@dev02> onmode -d secondary dev
```

The two servers will connect and after the Secondary clears its logical logs and receives all of the logical log records from the Primary the HDR setup is complete.

```
informix@dev02> onstat -m
```

```
IBM Informix Dynamic Server Version 14.10.UC7IE -- Updatable  
(Sec) -- Up 00:05:12 -- 1164976 Kbytes
```

```
Message Log File: /opt/informix-ids-14.10.UC7IE/tmp/online.log
```

```
13:42:09 Updates from secondary allowed
```

```
13:42:09 DR: Secondary server needs failure recovery
```

```
13:42:10 DR: Failure recovery from disk in progress ...
```

```
13:42:10 Logical Recovery Started.
```

```
13:42:10 10 recovery worker threads will be started.
```

```
13:42:10 Start Logical Recovery - Start Log 10, End Log ?
```

```
13:42:10 Starting Log Position - 10 0x1816018
```

```
13:42:10 Clearing the physical and logical logs has started
```

```
13:42:46 Cleared 3059 MB of the physical and logical logs in  
36 seconds
```

```
13:42:48 Started processing open transactions on secondary  
during startup
```

```
13:42:48 Finished processing open transactions on secondary  
during startup.
```

```
13:42:48 DR: HDR secondary server operational
```

```
13:42:49 B-tree scanners disabled.
```

```
13:42:50 Checkpoint Completed: duration was 0 seconds.
```

```
13:42:50 Tue Aug 3 - logunig 10, logpos 0x181e018,  
timestamp: 0x4a7af Interval: 723
```

```
13:42:50 Maximum server connections 0
```

```
13:42:50 Checkpoint Statistics - Avg. Txn Block Time 0.000, #  
Txns blocked 0, Plog used 14, Llog used 0
```

You really don't have to do anything else from this point forward to administer HDR, just sit back and relax. Your data is safer now.

## What do I do when the Secondary Server Fails?

If the Secondary fails and the logical log that was current at the time of the failure has not been reused (they're circular, remember) on the Primary then you can simply

restart the Secondary and it will automatically resync.

```
informix@dev02> oninit
informix@dev02> tail -40 $INFORMIXDIR/tmp/online.log
14:46:39 DR: ENCRYPT_HDR is 0 (HDR encryption Disabled)
14:46:39 Event notification facility epoll enabled.
14:46:39 IBM Informix Dynamic Server Version 14.10.UC7IE
Software Serial Number AAA#B000000
14:46:40 IBM Informix Dynamic Server Initialized -- Shared
Memory Initialized.

14:46:40 Started 1 B-tree scanners.
14:46:40 B-tree scanner threshold set at 5000.
14:46:40 B-tree scanner range scan size set to -1.
14:46:40 B-tree scanner ALICE mode set to 6.
14:46:40 B-tree scanner index compression level set to med.
14:46:40 Physical Recovery Started at Page (1:5623).
14:46:40 Physical Recovery Complete: 0 Pages Examined, 0
Pages Restored.
14:46:40 DR: Trying to connect to primary server = dev_hdr
14:46:41 Dataskip is now OFF for all dbspaces
14:46:41 Restartable Restore has been ENABLED
14:46:41 Recovery Mode
14:46:45 DR: Secondary server connected
14:46:46 Updates from secondary allowed
14:46:46 Updates from secondary allowed
14:46:46 DR: Using default behavior of failure-recovering
Secondary server

14:46:47 DR: Failure recovery from disk in progress ...
14:46:47 Logical Recovery Started.
14:46:47 10 recovery worker threads will be started.
14:46:47 Start Logical Recovery - Start Log 10, End Log ?
14:46:47 Starting Log Position - 10 0x182e018
14:46:48 Started processing open transactions on secondary
during startup
14:46:48 Finished processing open transactions on secondary
during startup.
14:46:48 DR: HDR secondary server operational
14:46:49 Logical Log 10 Complete, timestamp: 0x4a92d.
14:46:50 Logical Log 11 Complete, timestamp: 0x4a944.
14:46:51 Logical Log 12 Complete, timestamp: 0x4a975.
14:46:52 Logical Log 13 Complete, timestamp: 0x4a987.
14:46:54 B-tree scanners disabled.
14:46:55 Checkpoint Completed: duration was 0 seconds.
14:46:55 Tue Aug 3 - logunq 14, logpos 0x9018, timestamp:
0x4a9a4 Interval: 729

14:46:55 Maximum server connections 0
14:46:55 Checkpoint Statistics - Avg. Txn Block Time 0.000, #
Txns blocked 0, Plog used 15, Llog used 0
```

If your Secondary has been down for a while and the logical logs have rolled over there are 2 ways to recover. The easy way and the hard way.

The easy way is to reinitialize HDR by restoring the Primary to the Secondary again and running onmode -d secondary dev\_hdr on the Secondary.

The hard way is to restart the Secondary and when you see this message in the online.log

15:03:21 DR: Start failure recovery from tape ...

You can perform a Logical Restore to the Secondary using the logical log backups from the Primary. If you're backing up to a directory, copy the necessary logical log backups from the Primary to the Secondary, rename each backup to include the Secondary server name and use `ontape -l -d` to perform a Logical Restore.

```
informix@dev02> scp dev:/home/informix/backup/1log/* .
dev_0_Log0000000008
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000009
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000010
100% 1440KB 1.4MB/s 00:00
dev_0_Log0000000011
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000012
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000013
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000014
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000015
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000016
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000017
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000018
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000019
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000020
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000021
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000022
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000023
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000024
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000025
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000026
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000027
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000028
100% 96KB 96.0KB/s 00:00
dev_0_Log0000000029
100% 96KB 96.0KB/s 00:00
```

```

dev_0_Log0000000030
100% 96KB 96.0KB/s 00:00

informix@dev02> script_i_made_to_rename_the_files.ksh
informix@dev02> ls -l /home/informix/backup/llog
total 3648
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000008
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000009
-rw-rw---- 1 informix informix 1474560 Aug 3 14:56
dev02_0_Log0000000010
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000011
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000012
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000013
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000014
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000015
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000016
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000017
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000018
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000019
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000020
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000021
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000022
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000023
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000024
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000025
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000026
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000027
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000028
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000029
-rw-rw---- 1 informix informix 98304 Aug 3 14:56
dev02_0_Log0000000030

informix@dev02> ontape -l -d
Roll forward should start with log number 14
Restore is using file
/home/informix/backup/llog/dev02_0_Log0000000014 ...

```

```
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000014 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000015 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000016 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000017 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000018 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000019 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000020 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000021 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000022 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000023 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000024 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000025 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000026 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000027 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000028 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000029 ...  
Using the backup and restore filter /bin/gunzip.  
Rollforward log file  
/home/informix/backup/llog/dev02_0_Log0000000030 ...
```

Program over.

```
informix@dev02> tail -46 $INFORMIXDIR/tmp/online.log  
15:03:21 DR: Start failure recovery from tape ...  
15:03:28 Logical Recovery Started.
```



```

15:03:28 10 recovery worker threads will be started.
15:03:28 Start Logical Recovery - Start Log 14, End Log ?
15:03:28 Starting Log Position - 14 0x9018
15:03:29 Started processing open transactions on secondary
during startup
15:03:29 Finished processing open transactions on secondary
during startup.
15:03:29 DR: HDR secondary server operational
15:03:29 Logical Log 14 Complete, timestamp: 0x4a9e2.
15:03:29 Logical Log 15 Complete, timestamp: 0x4a9f9.
15:03:29 Logical Log 16 Complete, timestamp: 0x4aa0b.
15:03:29 Logical Log 17 Complete, timestamp: 0x4aa0b.
15:03:29 Logical Log 18 Complete, timestamp: 0x4aa33.
15:03:29 Logical Log 19 Complete, timestamp: 0x4aa45.
15:03:29 Logical Log 20 Complete, timestamp: 0x4aa45.
15:03:29 Logical Log 21 Complete, timestamp: 0x4aa6a.
15:03:29 Logical Log 22 Complete, timestamp: 0x4aa6a.
15:03:29 Logical Log 23 Complete, timestamp: 0x4aa94.
15:03:29 Logical Log 24 Complete, timestamp: 0x4aaa6.
15:03:29 Logical Log 25 Complete, timestamp: 0x4aaa6.
15:03:29 Logical Log 26 Complete, timestamp: 0x4aace.
15:03:29 Logical Log 27 Complete, timestamp: 0x4aace.
15:03:29 Logical Log 28 Complete, timestamp: 0x4aaf2.
15:03:29 Checkpoint Completed: duration was 0 seconds.
15:03:29 Tue Aug 3 - logunIQ 29, logpos 0x18, timestamp:
0x4aafc Interval: 730

15:03:29 Maximum server connections 0
15:03:29 Checkpoint Statistics - Avg. Txn Block Time 0.000, #
Txns blocked 0, Plog used 16, Llog used 0

15:03:30 Logical Log 29 Complete, timestamp: 0x4ab1f.
15:03:33 DR: Failure recovery from disk in progress ...
15:03:33 Logical Log 30 Complete, timestamp: 0x4ae61.
15:03:33 Checkpoint Completed: duration was 0 seconds.
15:03:33 Tue Aug 3 - logunIQ 31, logpos 0x15018, timestamp:
0x4aea4 Interval: 731

15:03:33 Maximum server connections 0
15:03:33 Checkpoint Statistics - Avg. Txn Block Time 0.000, #
Txns blocked 0, Plog used 13, Llog used 0

15:03:33 Checkpoint Completed: duration was 0 seconds.
15:03:33 Tue Aug 3 - logunIQ 31, logpos 0x17018, timestamp:
0x4aeaa Interval: 732

15:03:33 Maximum server connections 0
15:03:33 Checkpoint Statistics - Avg. Txn Block Time 0.000, #
Txns blocked 0, Plog used 0, Llog used 0

15:03:35 B-tree scanners disabled.
15:03:36 Checkpoint Completed: duration was 0 seconds.
15:03:36 Tue Aug 3 - logunIQ 31, logpos 0x20018, timestamp:
0x4aed2 Interval: 733

```



## What do I do when the Primary Fails?

When your Primary fails you can quickly make the Secondary server a Standalone (i.e. no HDR) server. Even if you have configured an Updatable Secondary you will need to do this since the writes on a Secondary are sent to the Primary under the covers.

Make the Secondary a Standalone server with the onmode -d standard command

```
informix@dev02> onmode -d standard
informix@dev02> onstat -m

IBM Informix Dynamic Server Version 14.10.UC7IE -- On-Line --
Up 00:30:32 -- 1164976 Kbytes

Message Log File: /opt/informix-ids-14.10.UC7IE/tmp/online.log
15:38:26 Logical Recovery Complete.
15:38:27 Quiescent Mode
15:38:27 Checkpoint Completed: duration was 0 seconds.
15:38:27 Tue Aug 3 - loguniq 31, logpos 0x4c018, timestamp:
0x4b0a6 Interval: 740

15:38:27 Maximum server connections 0
15:38:27 Checkpoint Statistics - Avg. Txn Block Time 0.000, #
Txns blocked 0, Plog used 0, Llog used 1

15:38:27 Started 1 B-tree scanners.
15:38:27 B-tree scanner threshold set at 5000.
15:38:27 B-tree scanner range scan size set to -1.
15:38:27 B-tree scanner ALICE mode set to 6.
15:38:27 B-tree scanner index compression level set to med.
15:38:27 DR: Reservation of the last logical log for log
backup turned on
15:38:27 SCHAPI: Started dbScheduler thread.
15:38:27 DR: new type = standard
15:38:27 Booting Language from module <>
15:38:27 Loading Module
15:38:27 SCHAPI: Started 2 dbworker threads.
15:38:28 On-Line Mode
```

When the old Primary is fixed and ready to be brought back online you have options.

Option 1 is to reinitialize HDR just like we did when setting up HDR for the first time. Except now dev02 will be the Primary and dev will be the Secondary. I like this option because it doesn't require any downtime.

Option 2 is to make dev the Primary again (easier to do if the logs have not rolled over on dev02.) This requires some downtime and assumes that the disks on dev were not the reason it went down and all of the data is still intact.

Switch dev02 to Quiescent Mode

```
informix@dev02> onmode -s
```

Change the HDR status of dev02 to Secondary

```
informix@dev02> onmode -d secondary dev_hdr
```

Start Informix on the Primary

```
informix@dev> oninit
```

If the logical logs have rolled over on the Secondary (while it was Standalone) you will need to do what we did before. Move the logical log backups that you need from dev02 to dev, change their names and run `ontape -l -d`

If everything works as advertised the Secondary will ship over the logs the Primary needs, they will be applied to the Primary and HDR will be restored.